University of Applied Sciences Mittweida and Chemnitz University of Technology at TRECVID Instance Search 2019

Rico Thomanek[1], Christian Roschke[1], Benny Platte[1], Tony Rolletschke[1], Tobias Schlosser[2], Manuel Heinzig[1], Danny Kowerko[2], Matthias Vodel[1], Frank Zimmer[1], Maximilian Eibl[2], and Marc Ritter[1]

[1]University of Applied Sciences Mittweida, D-09648 Mittweida, Germany
[2]Chemnitz University of Technology, D-09107 Chemnitz, Germany

Abstract. Identifying activities in large video collections remains a difficult process. (Partially) automated systems can be used to address different parts of these challenges. Object detection and classification are achieving ever higher detection rates using the latest state-of-the-art neural convolution networks. In our contribution to the task of *instance search* we specifically discuss the extension of a heterogeneous system that enables the identification performance for the recognition and localization of individuals and their activities by heuristically combining several state-of-the-art activity recognition, object recognition and classification frameworks. In our first approach to the task of *Instance Search.* (INS), which deals with the recognition of complex activities of persons or objects, we also integrate state-of-the-art neural network object recognition and classification frames to extract boundary frames from prominent regions or objects that can be used for further processing. However, basic tracking of objects detected by bounding boxes requires special algorithmic or feature-driven handling to include statistical correlations between frames. Our approach describes a simple yet powerful way to track objects across video images.

1 Introduction to our Appearance in the Instance Search Task

In 2019, we propose a novel approach for detecting and classify activity tasks in a cumulative process, consisting of specific components for automated evaluation and interactive search. In total, we analyzed more than 464 hours of video content with about 40 million frames. The overall problem is separated into several processing tasks for person recognition and dynamic activity recognition. Both recognition results are merged together to provide optimal results.

The focus of this paper deals with the basic software architecture and the flexible adaptation of the processing tasks to different application domains. The proposed system is extensible with different, docker-based modules for data preparation and / or data evaluation. In order to ensure a post-mortem-analysis of the results, all metadata information are stored and indexed.

We extracted data sets with more than 300 GB total size, which represents 464 hours of video footage, encoded in MPEG-4/H.264 format and separated into 244 video files from the TV series (*BBC EastEnders*).

Correspondence to: Marc Ritter
marc.ritter@hs-mittweida.de

2 System Architecture

Our system relies completely on a server-client approach in which a database server handles the persistent storage of all source and result data and the distributed cluster of a number of clients is responsible for the computation of recognition tasks. For efficient storage we use a distributed file system to access the stored raw data via HTTP, SCP and FTP. Like this, we circumvent saving raw data desktop computers, which would increase the administration overhead significantly. It is now possible to access the required source material in time while processing the data. A suitable application protocol is utilized for that purpose and also used to directly store results in the database. In addition, we have developed an API that can provide the data in an exchange format. Therefor, we implemented a session management layer to which we can ad or remove any number of processing nodes and services. The purpose of this session management layer is to scale and distribute all data processing tasks and to allocate resources to the processing nodes. This is needed to properly

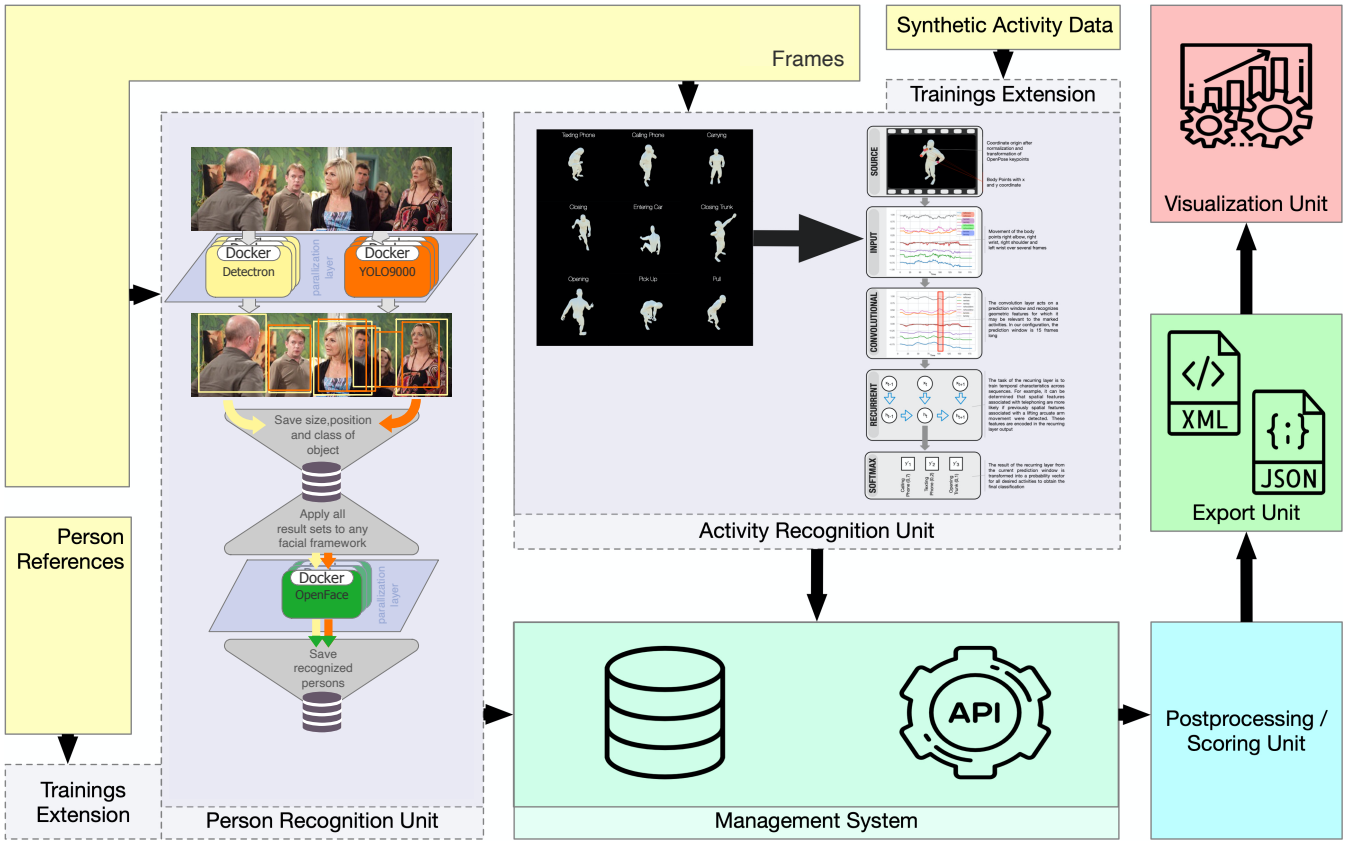

Figure 1: Our holistic system workflow for Instance Search.

execute the parallelization of processes and services. Automated data processing continues until all resources have finished successfully. Errors are handled by an error correction integrated in the session management layer. This triggers a restart of data processing. The pipeline is terminated when three unsuccessful attempts occur. The corresponding sets are marked as incorrect in the database and processing continues with the next block. All intermediate and final results generated in the entire process are stored in the database and can be accessed by each node directly using the corresponding API. All services are autonomously executed as docker containers. With this approach, we can distribute the client software to our hardware nodes without manual configuration. (NVIDIA)-Docker also allows the containers to be run multiple times using their automated resource management. The number of launched instances is adapted to the available CPU, RAM, and GPU on the host system.

To find instances in the Instance Search task, an open source frameworks for object instance identification and -tracking are used. The technical structure of our holistic system is shown in figure 1. Firstly, frames are extracted from our video data. We store those in our distributed file system reference them in the database. Then a keypoint recognition for multiple persons as well as an object recognition is performed based on the extracted frames. The resulting data is sent straight to the database where complementing transformed and normalized values are created from said person keypoints. Standardization of those keypoints is based on the distance between neck keypoint and the hip keypoint and transformed into a local separate coordinate system. The origin is placed at the neck keypoint. With coordinates based on this, the recognized objects and keypoints are stored in the database are transferred to the tracking algorithm separately. When we successfully track data over several frames, a unique ID is assigned. Resulting keypoints are then transferred to the LSTM for activity determination. To improve replicability of our results, we save all resulting data to our database. To solve the ever emerging problem of insufficient training data for our LSTM we developed a tool to generate synthetic human activities. Based on previously detected objects the activity classification is done by simple heuristics. Using SQL queries we deduce the activity and then merge the extracted partial results into a final data collection.

In order to perform the INS task, the system automatically transmits all results to the scoring unit. Internally, this consists of a business logic module that handles evaluation and creates a result object. We submit this to the export unit which structures it to an XML and/or JSON container. Subsequently, we use these containers to directly visualize the activity detection and tracking results via our custom visual-

ization unit. This allows us to develop an interface for the needed annotation of the data. Thorough analysis is conducted to evaluate and score the quality of our results obtained in the competitive period of TRECvid.

2.1 Frameworks

Various state-of-the-art frameworks can be used to identify people and objects. A disadvantage in this respect is that these frameworks only allow directory-based processing of images. For the integration of such frameworks into the existing infrastructure, the underlying source code has to be extended by online processing functions. Images that need to be processed are therefore no longer stored locally on the host system and are loaded directly from the central file system during runtime. A specially developed API enables concurrent access, processing and provision of the data.

In the described approach, all results are combined to characteristic vectors. Accordingly, the identification of an activity is achieved through the use of SQL queries. The relational linking of all results with the actual frame is done via the frame ID, which is realized via foreign keys. The following frameworks are used for a well-founded feature extraction: *Turi Create*, *Detectron*, *Yolo9000* and *OpenPose*. odo **OpenPose** is a framework for the real-time recognition of person keypoints, developed by Gines Hidalgo, Zhe Cao, Tomas Simon, Shih-En Wei Hanbyul Joo and Yaser Sheikh. Basic technologies of this framework are PyTorch, Keras or Tensorflow. Basically, the human body is recognized and key points are identified in individual images. The computing power of the system when determining body key points is invariant to the number of persons captured in the image. The CMU Panoptic Studio data set is used, in which 480 synchronized video streams of several persons involved in social activities are recorded and labeled, time-variable 3D structures of anatomical person key points in space are exported. The main functionality of the library is to assess and interpret the posture of individuals using 15 or 18 key points. OpenPose also provides the ability to estimate and render 2x21 hand key points. All functions are provided via an easy-to-use wrapper class.

TuriCreate is an open source framework for machine learning provided by Apple in 2017. The framework provides multiple libraries for handling different heterogeneous machine learning challenges. These include e.g. activity determination based on sensor values, image classification, image similarity, sound classification, object detection, clustering or regression (Sridhar et al., 2018). Our use case for this framework deals with activity detection. We use the library to create an activity classifier using the keypoints provided by OpenPose as normalized and transformed sensor data.

A framework for object recognition is **Detectron**. It was developed by Facebook, is based on the Deep Learning Framework Caffe2 and offers a high quality and powerful source code collection. When using Detectron, the height and width of the box are saved in addition to the x and y coordinates of the start point and the object classifications. We also use ResNet152 as a backbone net in combination with Mask R-CNN. The classes assigned to each image are taken from the Coco dataset (Lin et al., 2014).

YOLO is a network for real-time classification and localization of objects within an image. In contrast to a stack-based and step-by-step approach to object detection, object recognition is processed as a single regression problem. (Redmon et al., 2016)

YOLO9000 represents a revision of the YOLO framework, since it made significant mistakes, especially in object localization. Furthermore *YOLO9000* with over 9,000 different object categories was trained. (Redmon and Farhadi, 2016) We use *YOLO9000* to detect objects and their bounding boxes. The position and size of the bounding box is described by *YOLO9000* using upper left corner coordinates and lower right corner coordinates. The detected object class and the bounding box values are stored in the database.

2.2 Person detection and recognition

To create the required ground truth, we use Google's image search with the help of the software "google-image-download", which is licensed under MIT. Depending on the frequency of occurrence of a single actor, an average of 500 images can be collected with this method. Since only the faces are relevant in the context of the creation of a classifier, it is necessary to extract them from the downloaded images. *Face Recognition* allows to cut out faces and classify them. According to the requirements documentation of the framework *OpenFace* a minimum number of 10 images per person is required to create a classifier (numImagesThreshold=10). Accordingly, the frameworks *Face Recognition* and *FaceNet* require a minimum number of images somewhere in the single-digit range. With an average number of 150 images per person, we expect our soil truth data to exceed these requirements during the application period including a certain tolerance range.

The generated personal images are then indexed and made accessible via API. The person recognition consists of three steps. In the first step, all persons and their absolute positions are determined using the frameworks *YOLO9000*, *Detectron* and *Face Recognition*. For each framework, the recognition results are stored in a separate database table, with each framework recognizing a different number of people. In the second step, the framework *OpenFace* was used for person recognition. For this framework a classifier is generated based on the previously created Ground Truth. The resulting three tables are merged into the formula (1) using our heuristic evaluation scheme. In our approach, the recognition performance of each framework is considered equally. This means that the prediction value (2) for a recognized person can be evaluated by each framework with a maximum of 100 points. After the summation of all individual prediction

values, a maximum of 300 points (4) can be achieved per person.

$$predPerson = \sum_{h=0}^{r} (x_h \in K), \tag{1}$$

$$K = \{x \mid 0 \leq x \leq 100\}, \tag{2}$$

$$predPerson \in L, \tag{3}$$

$$L = \{y \mid 0 \leq y \leq 300\}. \tag{4}$$

For the subjective evaluation of the results, we used the web service already used in 2018. As shown in Figure 2, the number of error detections decreases with increasing score value, whereby the number of images is also reduced. According to the rules of the evaluation campaign the findings from the visual processing were not included in the automatic evaluation.

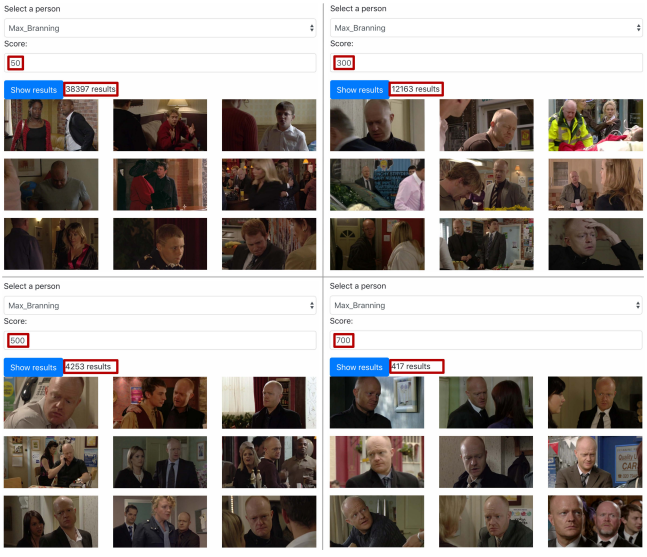

Figure 2: With a higher score value, the probability of person recognition increases.

2.3 Activity recognition

We have used different approaches to determine the activities we are looking for. These include the use of an audio classifier, the frameworks "Detectron" and "YOLO9000", a self-trained object classifier and a self-developed activity classifier based on the results obtained with "OpenPose".

To generate a suitable ground truth for the activity classifier, we designed a "unity" based tool that made it possible to generate synthetic data for the activities we were looking for. Using this tool, we were able to simultaneously record human activity from 10 different perspectives. In addition, the animations could be easily varied, resulting in a variety of activity animations. A total of 1452 synthetic animations were generated and divided into 198122 frames. With the help of *OpenPose* the body key points for the animated characters were determined and integrated into our system environment. The body key points provided by "OpenPose" describe the positioning of the bone point within the analyzed image. In order to derive an activity from the point movement occurring over several frames, it is necessary to convert these coordinates into a body-centered point. For this purpose we have chosen the neck as the origin of the coordinates. To perform the transformation, all body points belonging to a person are reduced by the x- and y-coordinates of the neck point. Another problem is the image resolution of the source material, because the determined "OpenPose" results relate to it. To create a dimensionless activity classifier, all body points must also be normalized. We have defined the normalization range as the distance between the neck and hips and scaled all other body points to the corresponding value range. Based on the normalized and transformed body keypoints, the activity classifier was created with "Turi Create". Each body point is considered as a two-dimensional sensor. The COCO model we use with *OpenPose* provides 18 body keypoints. Since each key point is described by an x and y coordinate, we get 36 individual sensor values. The activity classifier of *Turi Create* expects these sensor values assigned to an activity to be sorted chronologically in ascending order.

The object classifier was trained with 9504 images showing the activities drinking, eating, hodingBaby, holdingGlass, holdingPhone, hugging, laughing and kissing. These images were obtained using Google Downloader and manually labelled using "RectLabel" software. The object recognition has the task to classify and localize the activities displayed in the image. We used the "TuriCreate" framework to create the object classifier. The underlying model is an implementation based on TinyYOLO (YOLOv2 with a Darknet base network).

To derive specific activities, we also combined the objects detected by "Detectron" and "YOLO9000" with the results of the activity, object and audio classifier. Based on this, the found activities were extracted with a corresponding probability value. This extraction was carried out using suitable SQL queries.

2.4 Fusion of the determined scoring values

The fusion was done only in the database. With suitable SQL statements we linked the different results of the framework. For this purpose we generated different SQL views, combining the framework results of the activity analysis with the framework results of the person detection. The activity was evaluated based on the available prediction values and the matching classification. Data records in which several frameworks have classified the same activity are thus assigned a higher score. The maximum number of data records for the SQL views was limited to 1000.

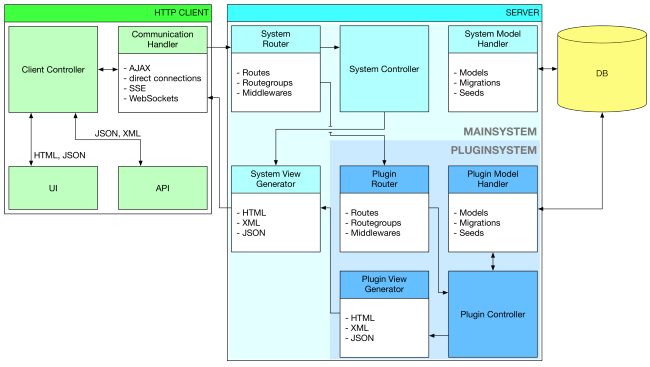

Figure 3: Management architecture

2.5 Simple Online and Realtime Tracking based multiple object tracking

The detection and tracking of multiple objects (MOT) in a sequence of videos constitutes one particular challenge human object interaction. As these tasks do not only include the identification of multiple objects within a wide range of different scenarios, they also encompass and substantiate the need for more complex estimation and compensation approaches when dealing with missing detections and occlusions. Therefore, a robust approach has to be leveraged to meet the ever-increasing demands of more complex problems and larger data sets.

2.5.1 MOT approaches

As current state of the art approaches to multiple object detection and tracking following *Xiu et al.* (2018) Xiu et al. (2018) and *Sun et al.* (2019) Sun et al. (2019) are more commonly developed as deep learning and deep neural network based systems, with or without any dedicated tracking – see also *Bergmann et a.* (2019) Bergmann et al. (2019) –, they are often computationally expensive and can not always be deployed in real-time. These often consist of multiple complex processing steps which are tailored to a specific application area or scenario, thus reducing their adaptability to novel application areas. Efficiently computable deep learning based approaches are also able to show state of the art results in current MOT challenges *Bewley et al.'s* (2016) Bewley et al. (2016).

To evaluate said approaches we conducted our tests on state of the art MOT challenges and benchmarks[1]. Our quality criteria are derived, among others, from the Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP) Bernardin and Stiefelhagen (2008), the ID F1 Score (IDF1) Ristani et al. (2016), the total number of identity switches (ID Sw.) Li et al. (2009), the total number of times a trajectory is fragmented (Frag), and, equally important, the processing speed (Hz) for all data sets.

[1] https://motchallenge.net

2.5.2 The Tracking algorithm

Our multiple object tracking system is based on a hybrid approach which benefits from both classical and deep learning based approaches. The system itself is in particular based on *Bewley et al.'s* (2016) Bewley et al. (2016)[2], *Wojke et al.'s* (2017) Wojke et al. (2017)[3], and *Wojke and Bewley's* (2018) Wojke and Bewley (2018)[4] work on multiple object tracking, which currently holds top scores within the Multiple Object Tracking Benchmark[5].

As candidate regions for tracking are given through the provided detections, the tracking itself is realized using Kalman filters. Each processed frame is then fed to generate a set of predictions for a given time frame for the next n seconds/frames, whereas all to this time processed frames with their respective detections are provided to update the system's current state until a given prediction confidence threshold is reached. To solve the mapping problem of the resulting bipartite graph (Fig. 4) from consecutive frames with two adjacent frames F_n and F_{n+1} with objects o_m $(n,m \in \mathbb{N}_{\geq 0})$, each object is compared to one another to generate a set of rankings which are then solved using the Hungarian method of linear programming. Finally, a matching cascade, consisting of a ResNet-based cosine distance as well as the Euclidean distance and IoU as metrics with nearest neighbor matching, is deployed to reach a final conclusion on the resulting object mapping. As a result, direction, speed, and motion vectors are not only estimated for the following frame, but also for all future frames within the given time window.

To reduce the number of ID Sw. and Frag, we extended the detection confidence and non-maxima suppression thresholds to dynamically correlate with the system's current state. We also introduce an additional motion compensation step to reduce the number of ID Sw. and Frag.

2.5.3 Postprocessing

To correlate the resulting tracks with the original detections of our preprocessing step, every interjacently appearing object is mapped accordingly as an estimation of our system to counteract missing detections as well as ID Sw. and Frag as investigated throughout testing.

Fig. 5 shows exemplary multiple object tracking results for our proposed system with the *EastEnders* and the *VIRAT* data set videos *5082189274976367100*, *VIRAT_S_000000*, *VIRAT_S_010000_00_000000_000165*, and *VIRAT_S_050000_01_000207_000361* (from top to bottom).

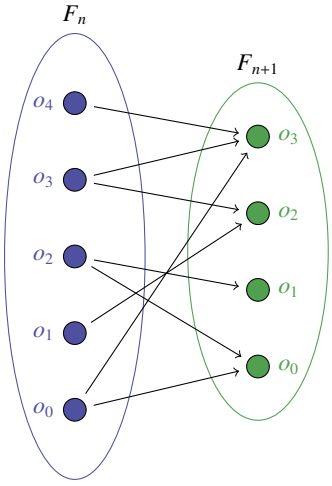

Figure 4: Illustration of a bipartite graph for object assignment between the two adjacent frames F_n and F_{n+1} with objects o_m $(n,m \in \mathbb{N}_{\geq 0})$.

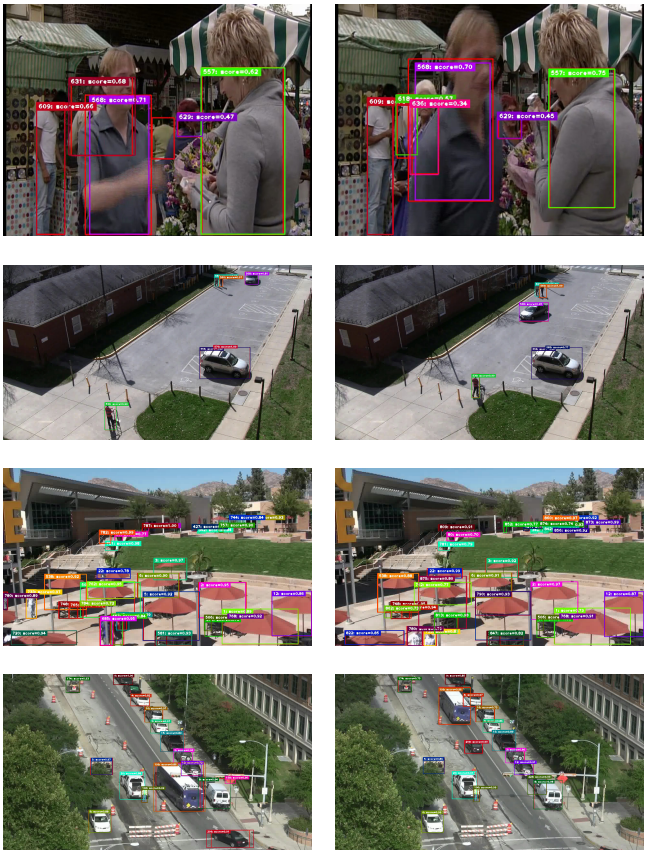

Figure 5: Exemplary multiple object tracking results for our proposed system with the *EastEnders* and the *VIRAT* data set videos *5082189274976367100*, *VIRAT_S_000000*, *VIRAT_S_010000_00_000000_000165*, and *VIRAT_S_050000_01_000207_000361* (from top to bottom).

2.6 System overview

Figure 3 shows our Laravel-based system architecture. A plugin manager has been added to the classic Laravel components. This allows the administration and integration of plugins and thus the expansion of the framework by additional functionalities. This allowed us to develop components for iBeacon communication and to add personalized content to the standard views.

The user interface for annotation developed in our appearance at TRECVid 2016 still allows to select a run and to visualize the current annotation status of the registered user. After the start of an annotation process a timer is generated and set to 5 minutes. The start time is transferred to the server and stored there to prevent fraud or manipulation of the timer. Several visualizations are generated for the user, each containing 2×3 result images of the corresponding query. The user can change the status of an image using keyboard shortcuts. Changing the status of an image changes the color to green and vice versa. After the timer has expired, the user interface is deactivated while the transfer is being uploaded to the server. Kahl et al. (2016)

In this period, we keep the basic interface and the underlying architecture whereas the business logic and the relations of objects has been integrated into the database with respect to performance issues. Furthermore, we add automated mechanisms to transfer the results of the automatic processing directly into the system. Thus we can significantly increase the speed of data preparation for intellectual annotation.

3 Results and Future Work in Instance Search

In this years iteration of the TRECVid evaluation we mainly use an improved and extended version of our system from last year. During development, we focused on high adaptability, which this year enabled us to participate in a wider range of TRECVid tasks and to file submissions for different, improved run setups than last year. With the refined approach, we were able to drop the interactive evaluation in favor of a broader variety of automatic configurations. Last year, this was not possible due to a lack in computational power. While we are still using the same hardware, parts of our system have significantly improved their performance. This can be seen as a rather big achievement, as it enables us to further optimize parametrization and system components on a functional level in the future.

For this year however, our system performs below average for most topics and categories when compared to the work of other teams. This is a fully expected outcome, as parts of our system are reused from last year, while newly

[2] https://github.com/abewley/sort
[3] https://github.com/nwojke/deep_sort
[4] https://github.com/nwojke/cosine_metric_learning
[5] https://motchallenge.net/tracker/DS_v2

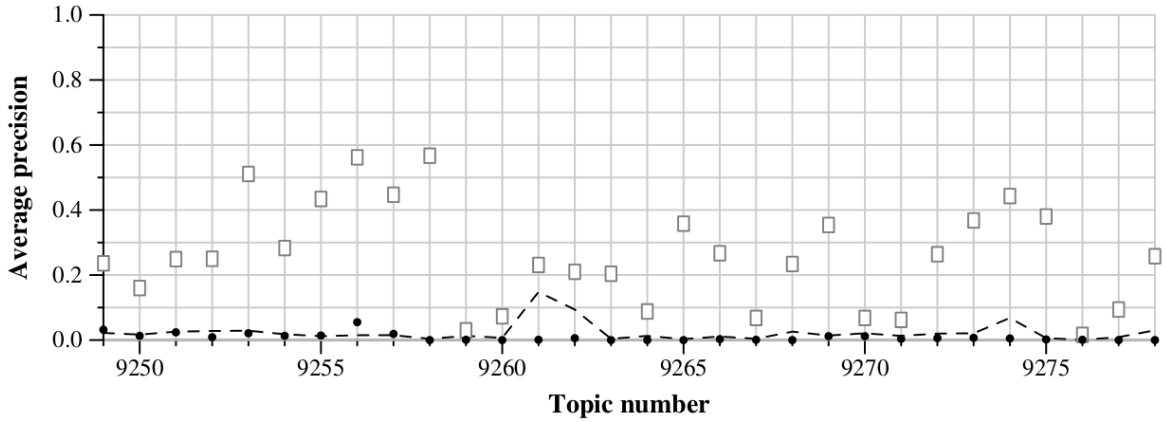

Figure 6: Results of our most elaborated run (Run 4) provided by the organizers. Dots are our scores, the dashed line indicates the median score and boxes represent the best results of other participants.

developed parts where primarily build with a focus on performance. A functional optimization has yet to be conducted and will most likely be in action next year. We are expecting a considerable increase in result quality after refinement. When compared directly to TRECVid 2018, present results of our four runs average on a significantly lower level. As we dropped our salient interactive runs, this is a coherent outcome. It is notable nonetheless, that the average result for our main runs show to be better as last year. The improvement itself is only mediocre, but given the fact that we did not rebuild a new system, but instead optimized intrinsic parameters, this fact proves our point for expectable better results through optimization of our core system in the next year. This is supported by the practical evidence, that our application is able to handle the provided data with acceptable computation times using our distributed approach that lets us calculate on various machines. We also improved the subsequent merging step in the central database. On an additional note, we are now capable of precalculating a every needed information from the given video corpus. With the acquired data, we are able to solve the retrieving step in under one second.

3.1 Run 1: Number of results across frameworks

With our first run, we were able to return 874 of 6,592 relevant shots at a mean average precision (MAP) of 0.008. The run scored a precision@total of 0.038. It features the full set of frameworks we assumed to be useful to fulfill the given task the best possible way. Framworks used are Yolo9000, Detectron, a self trained LSTM activity classifier, a self trained object detector and an audio classifier. FaceNet, FaceRecognition, Openface for person detection as well as Places365, Turic Create Similarity, Dominant Color, Yolo, Detectron for place recognition. We than prioritize the results according to person score, number of results over all frameworks and the probability of activity detection per shot. The best result is topic 9252.

3.2 Run 2: More than 1 person detection per shot

With nearly the same setup as Run 1, this automatic run features all frameworks that are present in our system. This time though, we introduce a heuristic to improve our false positive rate by eliminating all results with only one person detection in a whole shot. As a shot contains of many images, it is unplausible for a person to only appear in a single frame. Therefor, such detections are most likely false alarms of our system. This approach concludes with the returning of 931 relevant shots out of 6,592. However, this - in comparison to the first run slightly higher return count does not manifests itself in a better MAP (0.008). The precision@total even slightly drops to 0.037. A change in our best performing topic (changes to 9255) indicates the potential positive effects of such relatively simple heuristic approaches.

3.3 Run 3: Activity more than once detected

For our third automatic run, all available frameworks are used once more. This time, an heuristic that eliminates all shots with one time activity detections is used. Thoughts behind this are nearly the same as in Run 2: if any of the relevant activities is present in a shot, it will not be as short as 1-3 frames. Hence, shots with just one, very short activity are very likely to be false positives, which is why we eliminate them consequently. Using this setup, we end up with 939 retrieved relevant shots out of 6,592. With the resulting Mean Average Precision of 0.008 and a higher precision@total of 0.038 the run features identical results as Run 2. This is rather fascinating, as we use the same idea for two different heuristics and still end up with equal result characteristics. Again, topic 9255 proves to be the best scoring and comparison of the remaining topics also shows that their values are virtually the same. We conclude, that at least our systems frameworks for person and activity detection are prone to be fooled by equally characterized video material. Those

findings will be incorporated in the next iteration of our system.

3.4 Run 4: Activity and Person with more than 5% of shot length

For our last Run, we combine the former approaches and increase the strength of one main parameter. Detections will now only be taken into account, if they exceed a temporal span of 5% of the shots full length. This is our most harsh approach, as it filters out a substantial amount of detections in our system. In the end, this decision still turns out to be a good one. With a Mean Average Precision of 0.009 and a precision@total of 0.039 while finding 1014 relevant shots out of 6,592, this most complex and most restrictive Run emerges to be our best result in this years TRECVid evaluation period. The outstanding performance of topic 9526 also proves, that the functional combination of Run 2 and 3 with a strict mathematical boundary leads to a different outcome than the singled out approaches. This gives us a promising starting point for further investigation while improving our system with other combinations of heuristics and parametrization.

3.5 Conclusion

In this article it could be shown that the developed system could be extended by frameworks from the field of face and activity recognition. In this way the exploration of a new problem domain was possible. The research shows that the workflow developed in previous years is working and can be used as a methodical guideline. The modular structure of the workflow enables further investigations within the scope of automation. In addition, the results can be used to adapt further frameworks and improve them iteratively in the area of late fusion. In addition, it could be shown that the established infrastructure meets high requirements in the context of scalability and performance. We could notice that each framework delivers different prediction values even with a small change in the area of the image (a few pixels). The targeted multiple classification of the same object with slightly different image sections could possibly contribute to an increase in identification performance.

Acknowledgments

The European Union and the European Social Fund for Germany partially funded this research. This work was written in collaboration with the junior research group ”Agile Publika” funded by the European Social Fund (ESF) and the Free State of Saxony. It was also partially funded by the German Federal Ministry of Education and Research in the program of Entrepreneurial Regions InnoProfileTransfer in the project group localizeIT (funding code 03IPT608X). Program material in sections 1–3 is copyrighted by BBC. We want to thank all the organizers of these tasks, especially George Awad and Afzal Godil, for the hard work they put into the annotation, evaluation and organization of these challenges.

References

Bergmann, P., Meinhardt, T., and Leal-Taixe, L.: Tracking without bells and whistles, arXiv preprint arXiv:1903.05625, 2019.

Bernardin, K. and Stiefelhagen, R.: Evaluating multiple object tracking performance: the CLEAR MOT metrics, Journal on Image and Video Processing, 2008, 1, 2008.

Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B.: Simple online and realtime tracking, in: 2016 IEEE International Conference on Image Processing (ICIP), pp. 3464–3468, IEEE, 2016.

Kahl, S., Roschke, C., Rickert, M., Hussein, H., Manthey, R., Heinzig, M., and D. Kowerko, M. R.: Technische Universität Chemnitz at TRECVID Instance Search 2016, in: Proceedings of TRECVID Workshop, 2016.

Li, Y., Huang, C., and Nevatia, R.: Learning to associate: Hybridboosted multi-target tracker for crowded scene, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2953–2960, IEEE, 2009.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P.: Microsoft COCO: Common Objects in Context, ArXiv e-prints, 2014.

Redmon, J. and Farhadi, A.: YOLO9000: Better, Faster, Stronger, arXiv.org, p. arXiv:1612.08242, http://arxiv.org/abs/1612.08242v1, 2016.

Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A.: You Only Look Once - Unified, Real-Time Object Detection., CVPR, pp. 779–788, doi:10.1109/CVPR.2016.91, http://ieeexplore.ieee.org/document/7780460/, 2016.

Ristani, E., Solera, F., Zou, R., Cucchiara, R., and Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking, in: European Conference on Computer Vision, pp. 17–35, Springer, 2016.

Sridhar, K., Larsson, G., Nation, Z., Roseman, T., Chhabra, S., Giloh, I., de Oliveira Carvalho, E. F., Joshi, S., Jong, N., Idrissi, M., and Gnanachandran, A.: Turi Create, https://github.com/apple/turicreate, viewed: 2018-10-12, 2018.

Sun, S., Akhtar, N., Song, H., Mian, A. S., and Shah, M.: Deep affinity network for multiple object tracking, IEEE transactions on pattern analysis and machine intelligence, 2019.

Wojke, N. and Bewley, A.: Deep cosine metric learning for person re-identification, in: 2018 IEEE winter conference on applications of computer vision (WACV), pp. 748–756, IEEE, 2018.

Wojke, N., Bewley, A., and Paulus, D.: Simple online and realtime tracking with a deep association metric, in: 2017 IEEE International Conference on Image Processing (ICIP), pp. 3645–3649, IEEE, 2017.

Xiu, Y., Li, J., Wang, H., Fang, Y., and Lu, C.: Pose flow: Efficient online pose tracking, arXiv preprint arXiv:1802.00977, 2018.